PRINCIPI DI SISTEMI OPERATIVI

ESERCIZIO del 10 SETTEMBRE 2004

Un **museo** di arte contemporanea può contenere al massimo **V visitatori**. I visitatori arrivano in **gruppi** fino a **P** persone (con $1 \leq P \leq V$). I visitatori appartenenti allo stesso gruppo devono entrare tutti contemporaneamente. Il museo ha a disposizione **G guide**, che possono essere richieste dai gruppi. Ogni gruppo può richiedere al massimo una guida, oppure nessuna se preferisce visitare il museo da solo.

Si implementi una soluzione usando il costrutto monitor per modellare il **museo** e i processi per modellare i **gruppi** e si descriva la sincronizzazione tra i processi. Nella soluzione si massimizzi l'utilizzo delle risorse. Si discuta se la soluzione proposta può presentare starvation e in caso positivo per quali processi, e si propongano modifiche e/o aggiunte per evitare starvation.

```
program museo_arte_contemporanea

const     V = ...; { capacità del museo }
const     P = ...; { consistenza massima dei gruppi }
const     G = ...; { numero di guide }
type      per = 1..P; { numero di persone in un gruppo }

type gruppo = process (n: per; g: boolean)
begin
          m.entra(n, g);
          <visita il museo >
          m.esci(n, g);
end

type museo = monitor
{ variabili del monitor }
var  n_vis : integer;
     { numero di visitatori nel museo }
     n_guide : integer;
     { numero di guide libere }
     sospesi : integer;
     { numero di visitatori sospesi }
     coda : condition;
     { coda su cui sospendere i gruppi }

procedure entry entra(n: per; g: boolean)
begin
     while ((n + n_vis) > V) or (g and (n_guide = 0)
     do { se non c'è posto o se non c'è una guida libera }
     begin
          sospesi++;    coda.wait;     sospesi-- ;
     end
     n_vis := n_vis + n; { i visitatori entrano }
     if g then
          n_guide-- ;
end
```

```
procedure entry esci(n: per; g: boolean)
var s, i: integer;
begin
      n_vis := n_vis - n; { i visitatori escono dal museo }
      if g then
            n_guide++ ;
      s := sospesi; { risveglia tutti i visitatori in coda }
      for i := 1 to s do
            coda.signal;
end

begin { inizializzazione delle variabili }
      n_vis := 0;
      n_guide := G;
      sospesi := 0 ;
end

var  m: museo; { il nostro monitor }
      g1, g2, ... : gruppo (j, true|false);

begin end.
```

Starvation

La soluzione proposta può presentare starvation nei confronti dei gruppi numerosi.
Si può limitare utilizzando una coda per ogni dimensione possibile (con un array [1..P]) oppure utilizzando due code, una con priorità maggiore dell'altra dove sospendere i visitatori dopo alcuni tentativi di ingresso andati a vuoto.

Nota

La soluzione considera un visitatore singolo come un gruppo composto da una sola persona